Beyond Algorithmic Analysis

Assignment written by Katie Creel and Nick Bowman

Thus far in CS106B, you have been introduced to the technique of benchmarking (i.e., measuring run time) and saw in your first assignment how the choice of algorithm can have significant impacts on performance. At the start of Week 3, we are expanding and formalizing our accounting of algorithmic performance using a technique called Big-O analysis, which will inform our study of algorithms throughout the rest of the course. However, we want to approach the topic of efficiency and optimization with a certain amount of cautious skepticism in terms of its ethical implications. A company might argue that as computers become increasingly more powerful, an effort to make a program slightly more efficient or use a bit less memory does not warrant the investment of (expensive) programmer time, but is the corporate bottom line the only cost to consider? What about the savings of environmental and energy resources? (I'm looking at you, cryptocurrency mining, and the vast amount of squandered computer cycles) On the other hand, prioritizing efficiency at the cost of sacrifices in effectiveness, quality, or ease of use seems a poor tradeoff of a different sort. This short ethical reflection is designed to prepare you to learn about Big-O by pausing to consider some of these challenges.

References

> **Q8**. In a short paragraph, describe a real or plausible scenario in which reworking an algorithm to improve its efficiency might benefit Earth's environment or humanity as a whole. Include your thoughts on how a software engineer working on this piece of code might identify such potential benefits and take them into consideration when designing the code.

As ethical and socially-conscious computer programmers, we also know that many considerations other than the speed and memory use are important in choosing the appropriate algorithm for a particular use. Dr. Gillian Smith, an associate professor at the Worcester Polytechnic Institute, identifies an interesting fallacy that computer scientists often fall into when applying algorithmic analysis techniques like benchmarking and Big-O analysis:

If it's more efficient, it's better. The quality of a program, independent of how people interact with it, should be evaluated only with respect to how well it minimizes cost.

The following case study illustrates the importance of supplementing efficiency and performance analyses with human-centric evaluation.

In 2006 the state of Indiana awarded IBM a contract for more than $1 billion to modernize Indiana's welfare case management system and manage and process the State's applications for food stamps, Medicaid and other welfare benefits for its residents. The program sought to **increase efficiency and reduce fraud** by moving to an automated case management process. After only 19 months into the relationship, while still in the transition period, it became clear to Indiana that the relationship was not going as planned. In particular here are some "lowlights" of the system's failures to provide important and necessary services for those in need:

- "Applicants waited 20 or 30 minutes on hold, only to be denied benefits for 'failure to cooperate in establishing eligibility' if they were unable to receive a callback after having burned through their limited cellphone minutes."
- "Applicants faxed millions of pages of photocopied driver's licenses, Social Security cards, and other supporting documents to a processing center in Marion, Indiana; so many of the documents disappeared that advocates started calling it "the black hole in Marion" […] Any application missing just one of tens to hundreds of pieces of necessary information or paperwork were automatically denied."
- "By February 2008, the number of households receiving food stamps in Delaware County, which includes Muncie, Indiana, dropped more than 7 percent, though requests for food assistance had climbed 4 percent in Indiana overall." (Quotations from [Virginia Eubanks](#))

In light of these failures, the State of Indiana cancelled its contract with IBM and sued the

company for breach of contract, stating that the company had failed to deliver a system that helped people get the services they needed. In court, IBM argued that they were not responsible for issues related to wait times, appeals, wrongful denials, lost documents, etc. as the contract only stated that a successful system would succeed by reducing costs and fraud. IBM's system did reduce costs, but did so by denying people the benefits they needed. In light of this, we would like you to consider the following questions:

> **Q9**. According to the contract that IBM struck with the state of Indiana, the criteria for optimization were improving efficiency of the overall welfare system and reducing fraud. Criteria for reducing wait times and wrongful denials were not included. However, wrongfully denying benefits has a huge negative impact on the citizens who rely on the system. If criteria like minimizing wrongful denials were not included in the contract, should engineers have included them in their optimization algorithm? Why or why not?

> **Q10**. Imagine that after completing CS106B you are hired at IBM as an engineer working on this system. How might you have approached designing and setting the goals of this system? How might you apply algorithmic analysis tools to build a system that achieved the desired goals? Could you do so in a way that avoids the severe negative impacts on users of the system that are outlined in the case study?

References

If you are interested to learn more, we highly recommend reading more from Virginia Eubanks.

- Virginia Eubanks, *"Want to Cut Welfare? There's an App for That."*, The Nation, May 27, 2015. https://www.thenation.com/article/archive/want-cut-welfare-theres-app/
- Virginia Eubanks, *"Automating Inequality: How High-Tech Tools Profile, Police, and Punish the Poor"*, Macmillan, 2018. https://us.macmillan.com/books/9781250074317/automatinginequality